AMENDMENTS TO THE SPECIFICATION

Applicants note the original application was filed without paragraph numbers. To facilitate the incorporation of these amendments, Applicants have numbered the paragraphs. Please replace Paragraphs [0058], [0069], [0074], [0078-0079], and [0171] with the following paragraph rewritten in amendment format:

[0058] Figures 36-47 are comparative data charts indicating the performance of various configurations of the ~~preferred~~ embodiment of the present disclosure relative to other communication utilities.

[0069] The ~~preferred~~ embodiment of the present disclosure preferably uses three variations of the vcuSend() routine, with the particular vcuSend() routine being used depending upon the channel configuration of the channel involved in the data transfer. However, it should be noted that a vcuSend() routine that requires more information than another vcuSend() routine can preferably be substituted for that other vcuSend() routine.

[0074] To receive data, the ~~preferred~~ embodiment of the present disclosure preferably uses the vcuRecv() routine: int vcuRecv(int *channel*, char* *&data*, int *&dataSize*, int *flags*). The vcuRecv() routine operates to receive the next message waiting in the Rx queue for the channel specified in the argument, if such a message exists. The *flag* options for vcuRecv() are VCU_NO_BLOCK (which is the default setting) and VCU_BLOCK. When the flag is VCU_BLOCK, the vcuRecv() routine

blocks until the data arrives. However, it is worth noting that a timeout option can be used to end the block after the passage of a specified amount of time.

[0078] *Channel Configuration Types:*

The embodiment of the present disclosure preferably allows user to define (and redefine) the transmission characteristics of at least one communication channel, and more preferably, each communication channel. It is preferred that the user be given the ability to define (and redefine) aspects such as: the number of communication channels, the maximum size of a single data transfer for each channel, the conditions under which DMA is used for data transfers across the bus, and how each channel is to handle data transfers.

[0126] Lastly, the "step forward" button 238, the "step back" button 240, the "previous 32" button 242, the "next 32" button 244, the "jump to start" button 246, and the "jump to end" button 248 operate to correspondingly change the set of displayed channels. In ~~a preferred~~an embodiment, the GUI only displays 32 channels at a time, but can manage a much larger number of channels (preferably 4096). The "step forward" button and "step back" button operate to, respectively, increment or decrement the channels in the displayed set by one. For example, when step forward is selected while channels 1-32 are displayed, the resultant channels will be 2-33. The "previous 32" and "next 32" buttons operate to increment or decrement in units of 32. The "jump to start" button operates to display channels 1-32, while the "jump to end" button

operates to display the last set of channels. It is preferred that these buttons only be enabled when the number of channels exceeds 32.

[0171] While the present disclosure has been described above in relation to its ~~preferred~~ embodiment, various modifications may be made thereto that still fall within the disclosure's scope, as would be recognized by those of ordinary skill in the art.